DeeR-VLA: Dynamic Inference of Multimodal Large Language Models for Efficient Robot Execution

Yang Yue[*,1] Yulin Wang[*,1] Bingyi Kang[2] Yizeng Han[1] Shenzhi Wang[1]
Shiji Song[1] Jiashi Feng[2] Gao Huang[†,1]

[1] Department of Automation, BNRist, Tsinghua University [2] ByteDance

Presented by Yue Yang

yueyang22f@gmail.com

paper code

Background: MLLM for robot

LLM and multimodal LLM

GPT - 4

Llama-3.2

Gemini

Qwen2.5

Empowering embodied AI (EAI):

- 🤩 vision-language comprehension
- 🤩 problem-solving capabilities

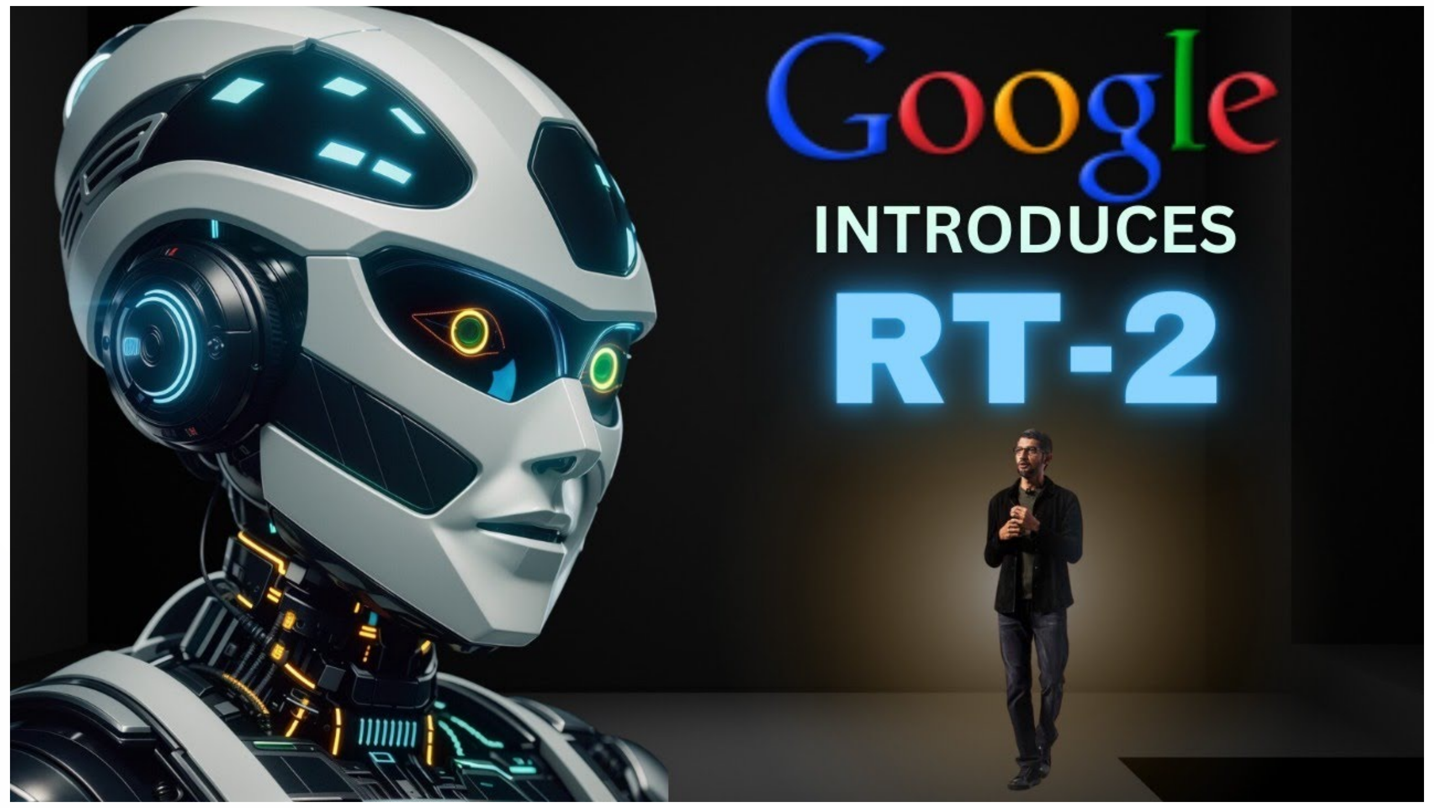
Generalist robot

Smart Home

Industrial Production

Elderly Care

Key Challenge: High Costs for Deployment



Embodied Large Model (ELM) **Google RT-2**

- **55B parameters**
- Memory: > **110G**
- Computation: > **200 GPUseconds** for a simple grasp task

EAI deployment

- High **Real-time** Requirements
- **Limited Memory** on Edge Devices (ORIN 8G/16G)
- Limited Onboard **Battery Capacity**

How to Reduce Costs in Embodied Large Model Inference?
Promote the Democratization for Research and Industrial

Motivation

Tsinghua University

Key Challenge: High computation/ memory costs for deployment

Solution: dynamic inference for Embodied Large Models

Embodied Large Models: **Static, Equal** Processing of All Task Instructions and Stages

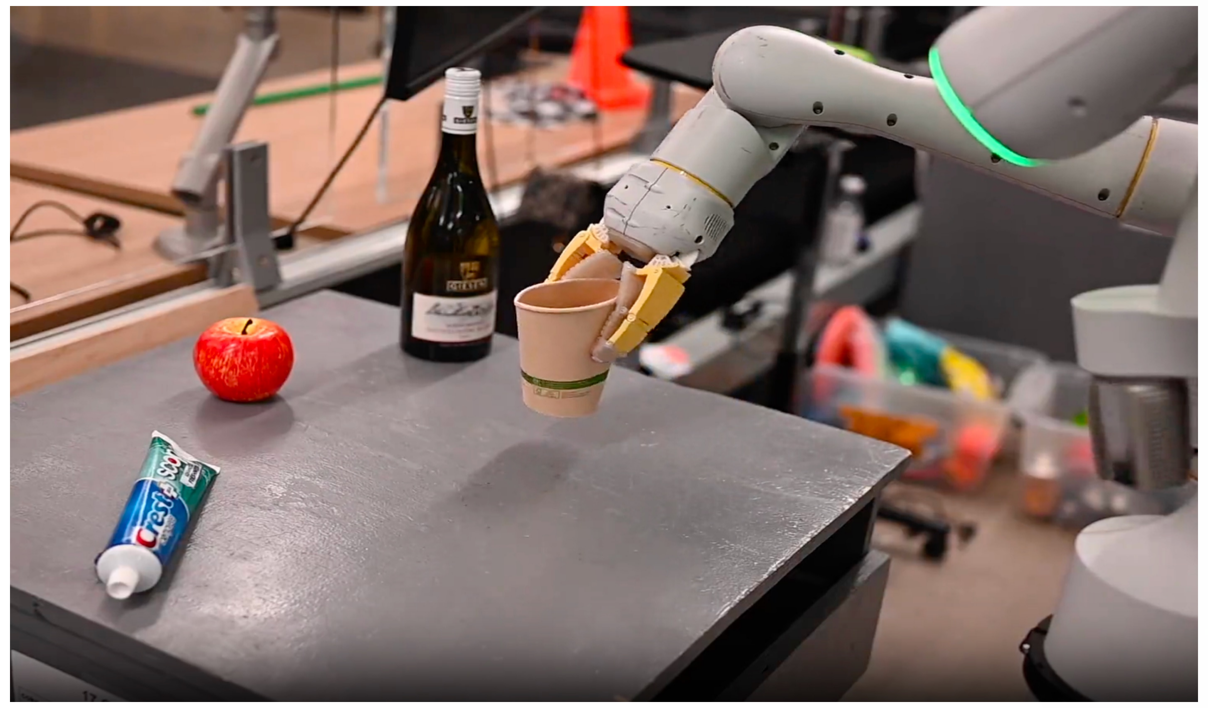
Human Decision Systems: **Dynamically and Adaptively** Adjust Costs Based on Task and Scenario

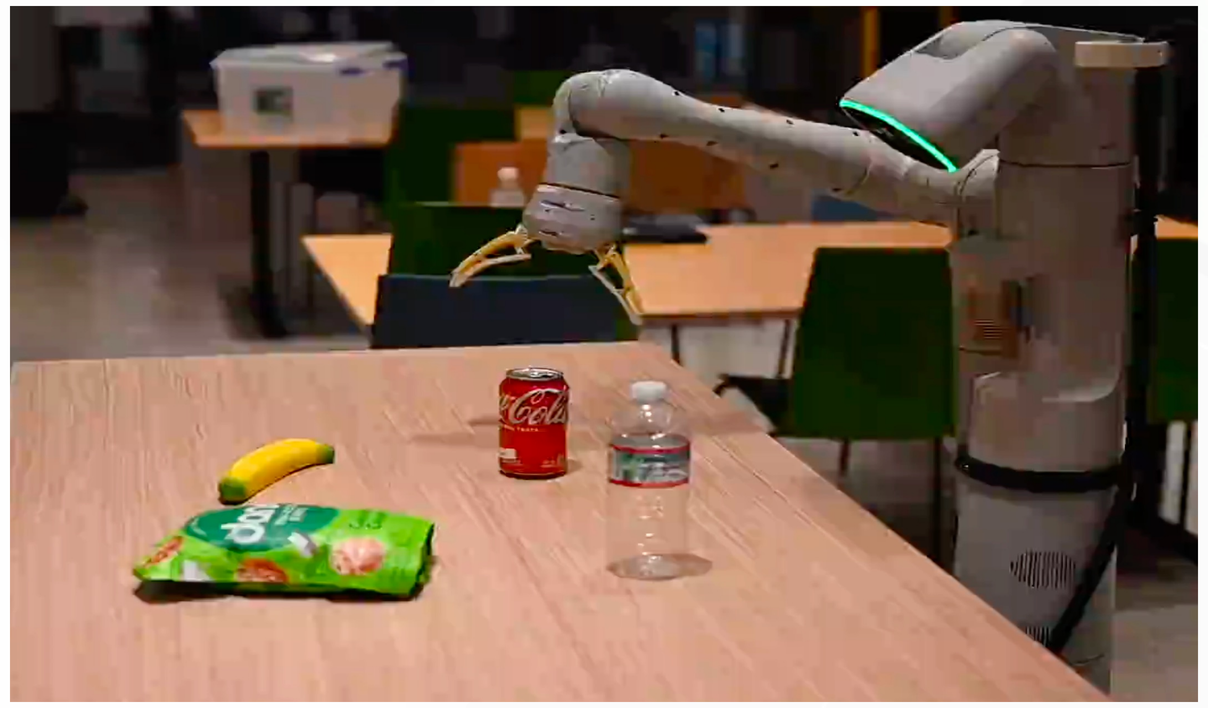
OpenAI o1: Allocate More Computational Resources for Complex Tasks

Tasks

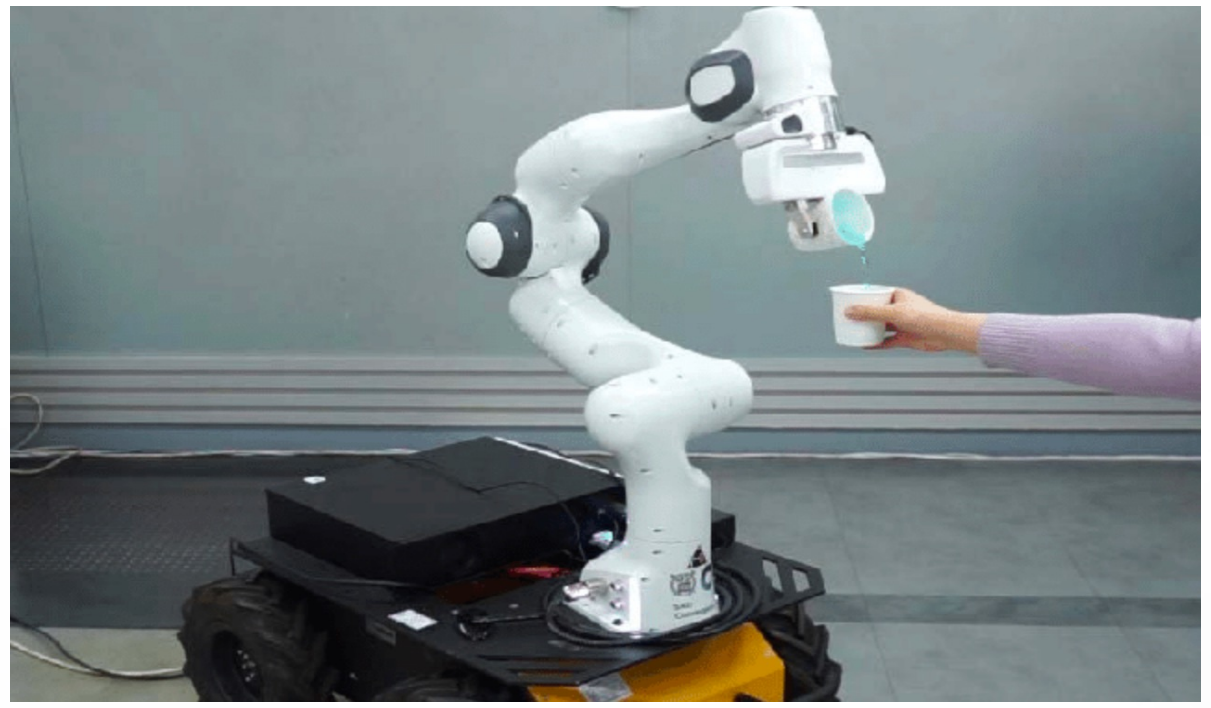
Stages

Easy





Hard



Method

- **Larger models** are redundant for easy circumstances, wasting computation
- Dynamic Early-Exit Framework for Robotic VLA Model (DeeR-VLA)
  - seek to **automatically configure the size of MLLMs** conditioned on situation
  - **Early-exit**: exit at shallow layers for easy tasks

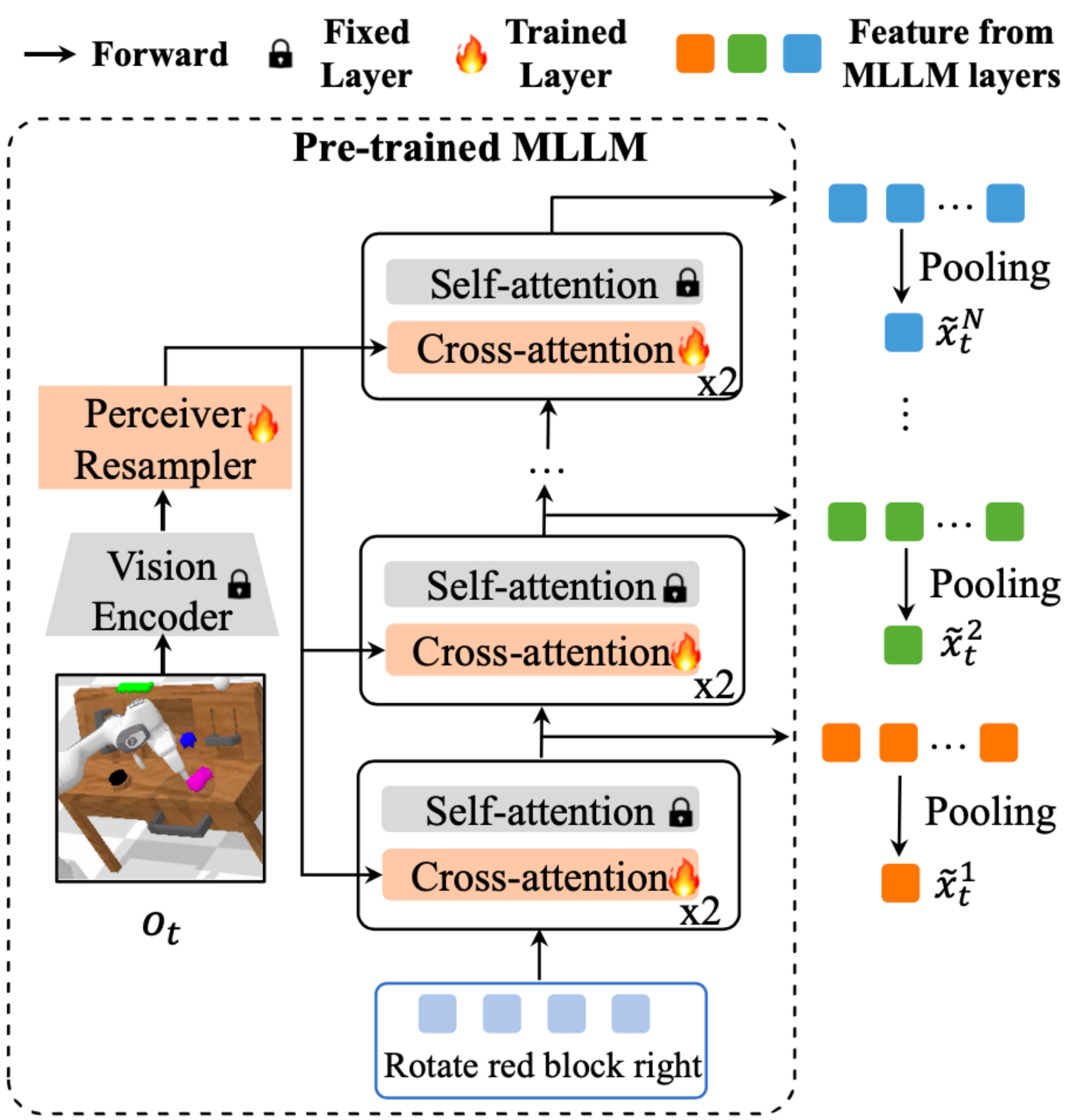
# LLM layers	24	12	6
GFLOPs/action (LLM)	31.2	15.6	7.8
Task success rate %	78.9	78.0	75.7

Method

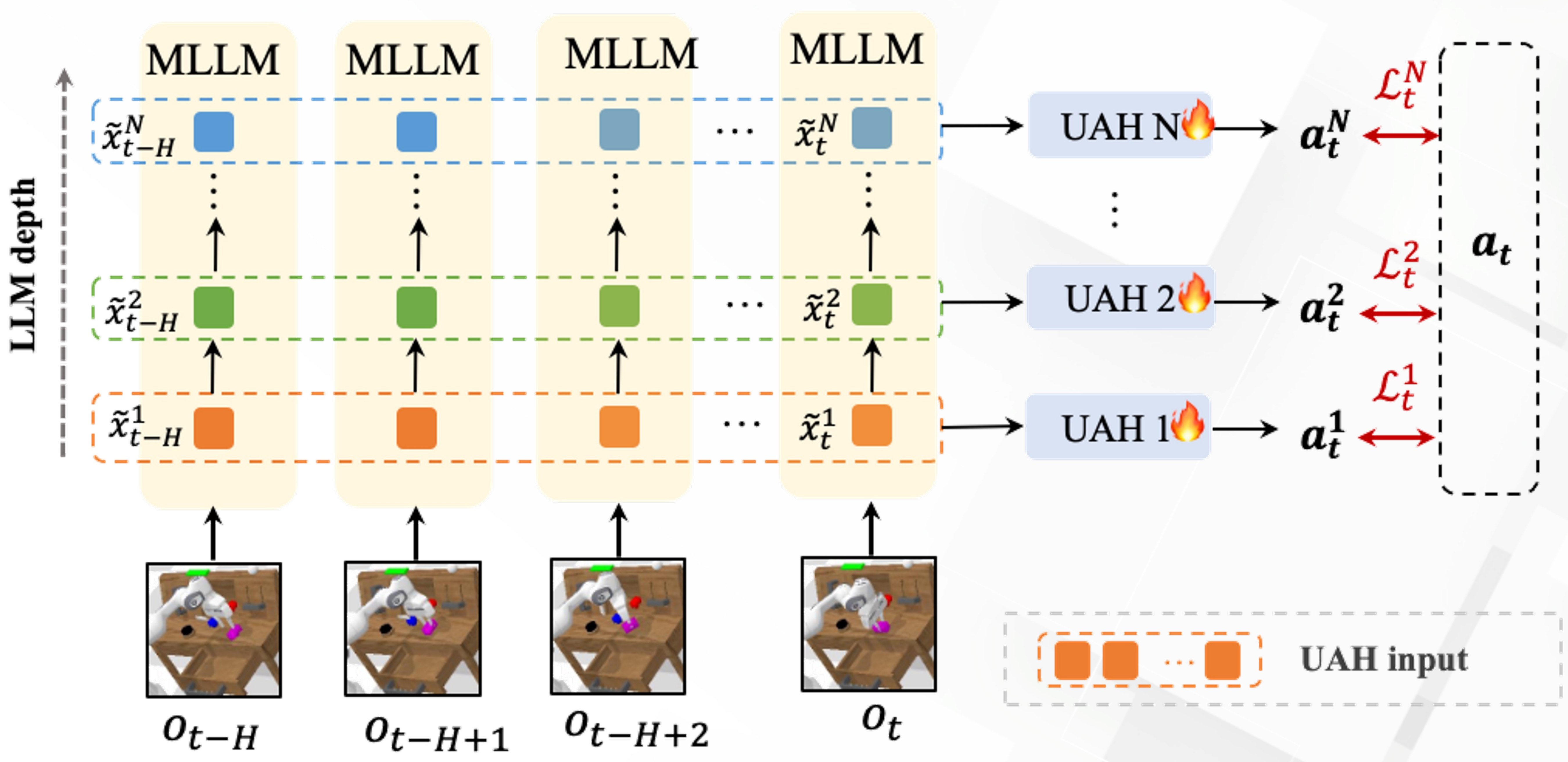
Dynamic Early-Exit Framework for Robotic VLA Model (DeeR-VLA)

➢ Dynamic Architecture of ELM

Dynamic-depth Network Architecture



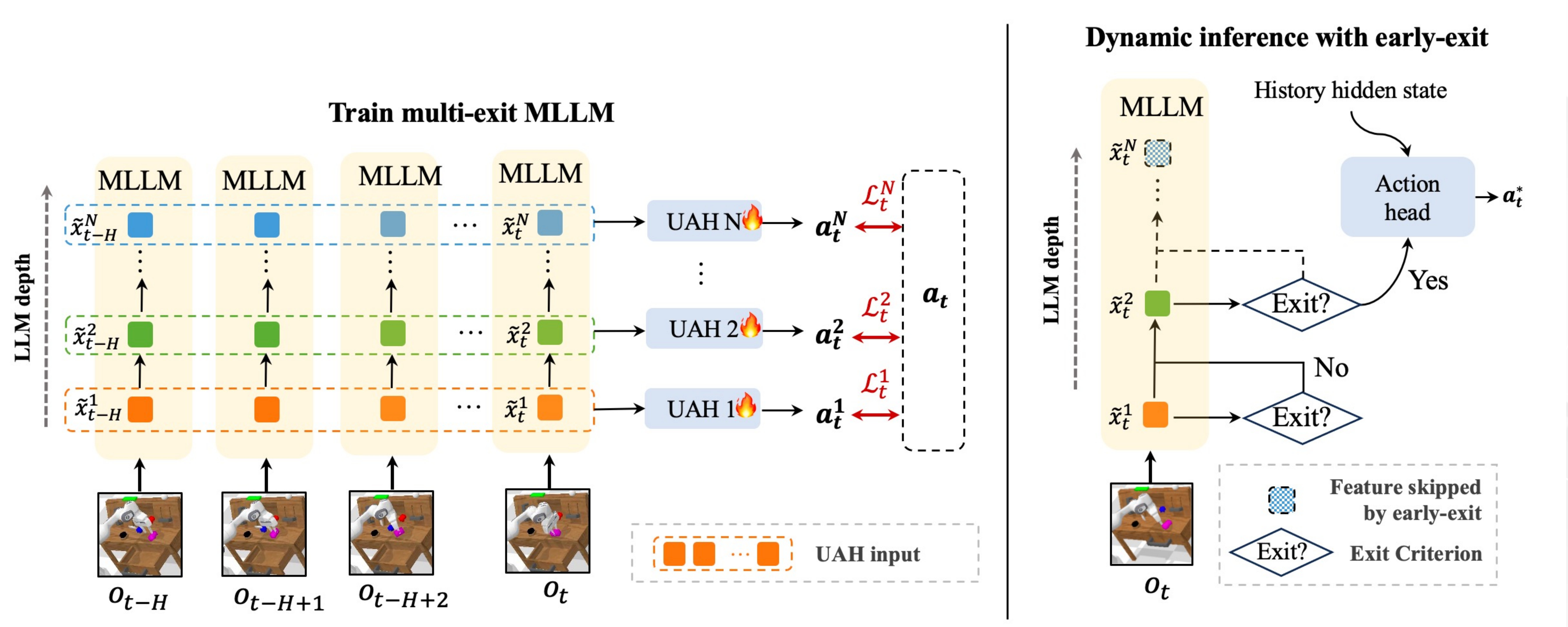
Train multi-exit MLLM



$$\mathcal{L}_{\text{aux}} = \sum_{j=1}^{N} \sum_{i=0}^{H-1} \mathcal{L}(a_{t+i}^j, a_{t+i})$$

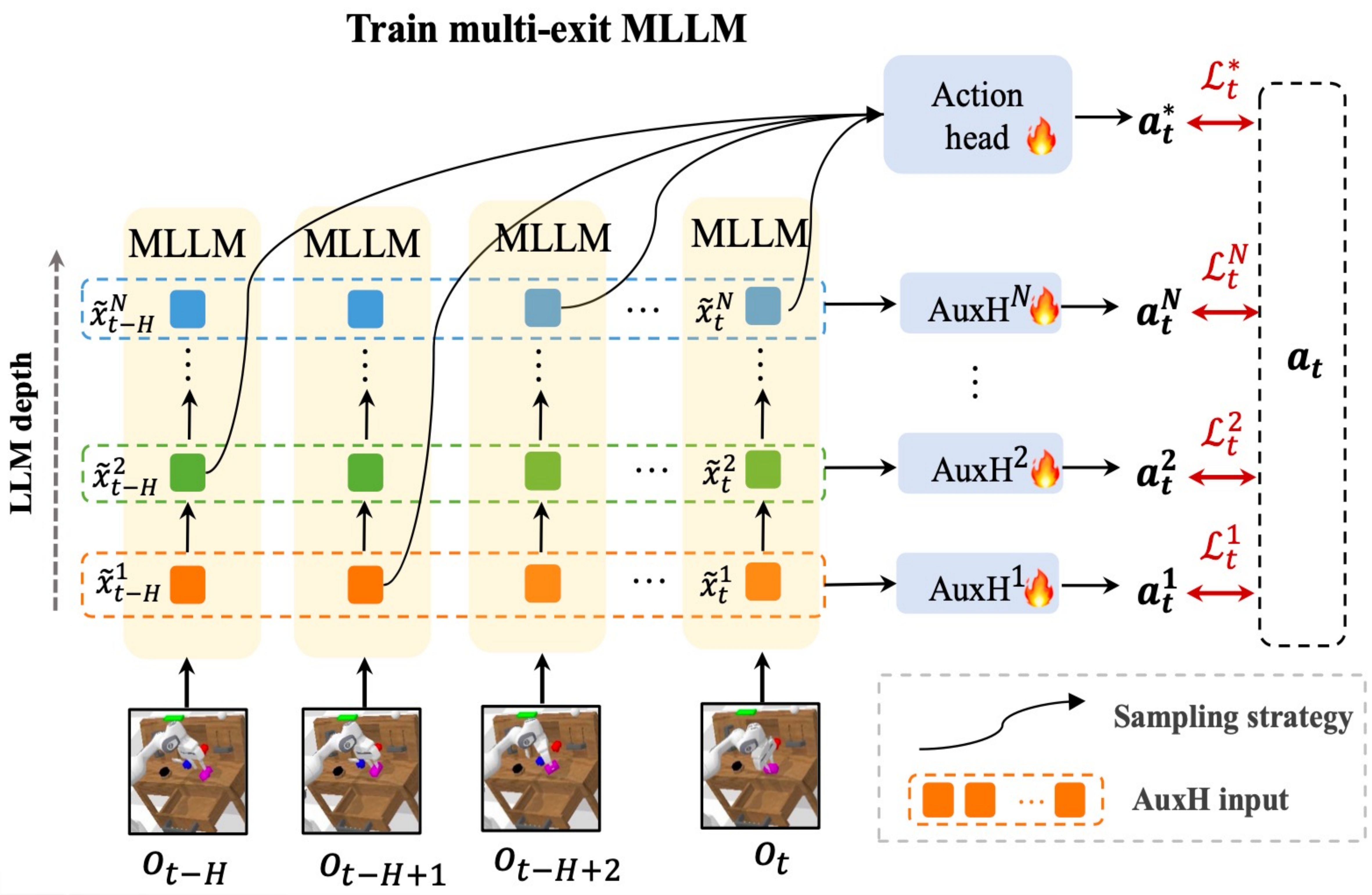
Method

- Dynamic Early-Exit Framework for Robotic VLA Model (DeeR-VLA)
  - Dynamic Architecture of ELM
  - **Feature distribution Gap between train and dynamic infer**
    - Train exit sequence: 2,2,2,2; 6,6,6,6; ...
    - Infer exit sequence: 1,3,3,5; 6,1,3,6, ...



Method

- **Dynamic Early-Exit Framework for Robotic VLA Model (DeeR-VLA)**
  - Dynamic Architecture of ELM
  - **Dynamic Feature Sampling Training Strategy**



Two Sampling Strategies s1, s2

- s1: uniform sample
- s2: random switch once in a window, e.g., 2,2,4,4; 6,6,6,1

$$\mathcal{L}_{\text{aux}} = \sum_{j=1}^{N} \sum_{i=0}^{H-1} \mathcal{L}(a_{t+i}^{j}, a_{t+i})$$

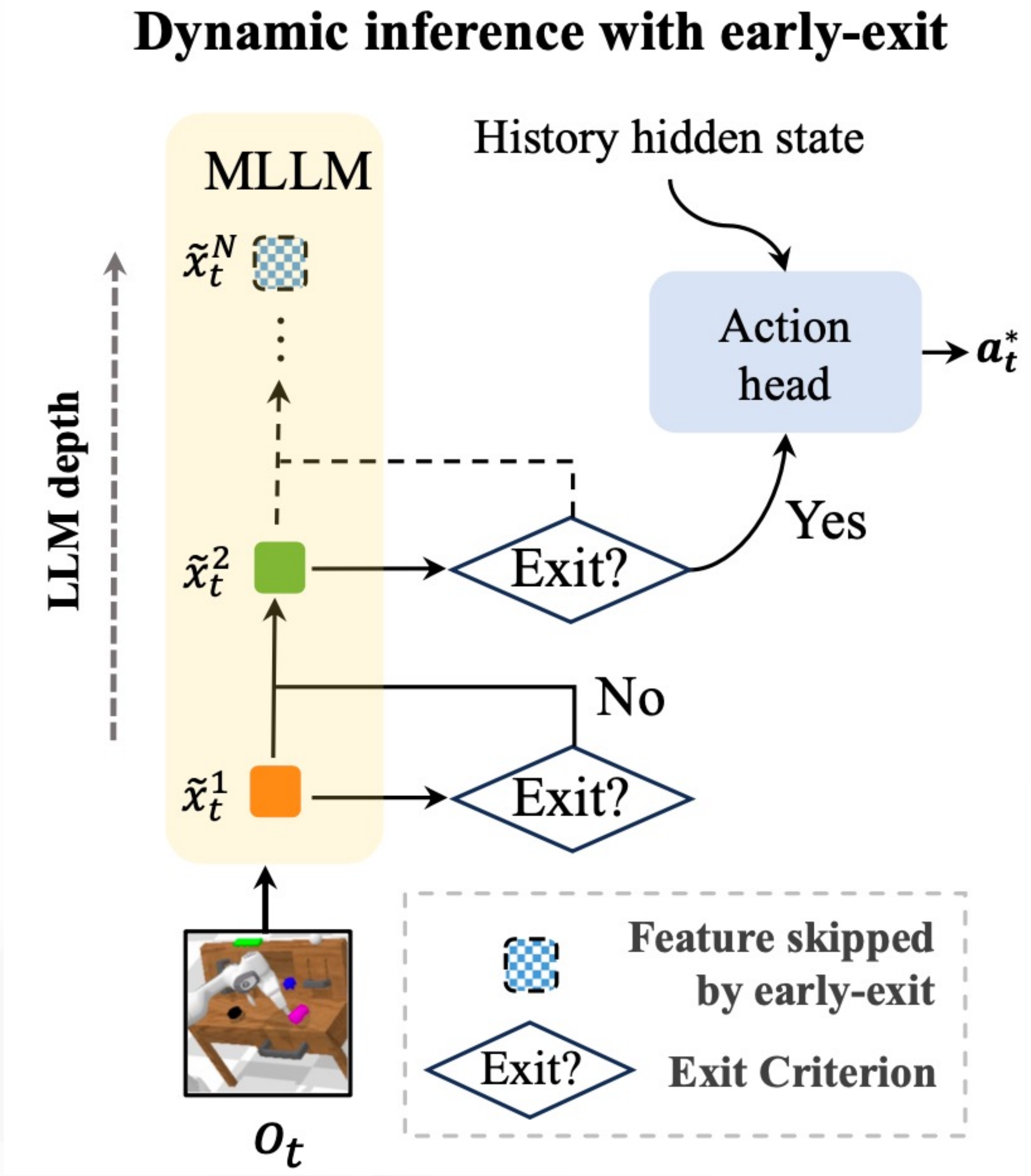
$$\mathcal{L}^* = \sum_{s \in \{s_1, s_2\}} \sum_{i=0}^{H-1} \mathcal{L}(a_{t+i}^{*,s}, a_{t+i})$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}^* + \mathcal{L}_{\text{aux}}$$

Method

Dynamic Early-Exit Framework for Robotic VLA Model (DeeR-VLA)

- Dynamic Architecture of ELM
- **Dynamic Inference for Embodied Tasks**



1. Dynamic Depth Selection Metric for Embodied Tasks:

Action Consistency

$$\|\pi_\theta(\tilde{x}_t^i, h_{t-1}) - \pi_\theta(\tilde{x}_t^{i-1}, h_{t-1})\|_2 < \eta_i,$$

2. Given Any Computational Cost and Memory, Solve for thresholds for each exit

$$\max_{\eta_1,\eta_2,\ldots} \mathrm{Scc}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}),$$

subject to

$$\mathrm{FLOPs}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) < B,$$
$$\mathrm{MFLOPs}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) < G,$$
$$\mathrm{Mem}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) < M.$$

Method

- Dynamic Early-Exit Framework for Robotic VLA Model (DeeR-VLA)
  - How to solve optimal threshold
    - (1) **default**: Solving problem using a demonstration dataset
    - (2) **beta**: Solving with *online interactions* (Bayesian Optimization)

$$f_{\text{obj}} = \text{Scc}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) - P$$

2. Given Any Computational Cost and Memory,
Solve for thresholds for each exit

$$\max_{\eta_1, \eta_2, \ldots} \text{Scc}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}),$$

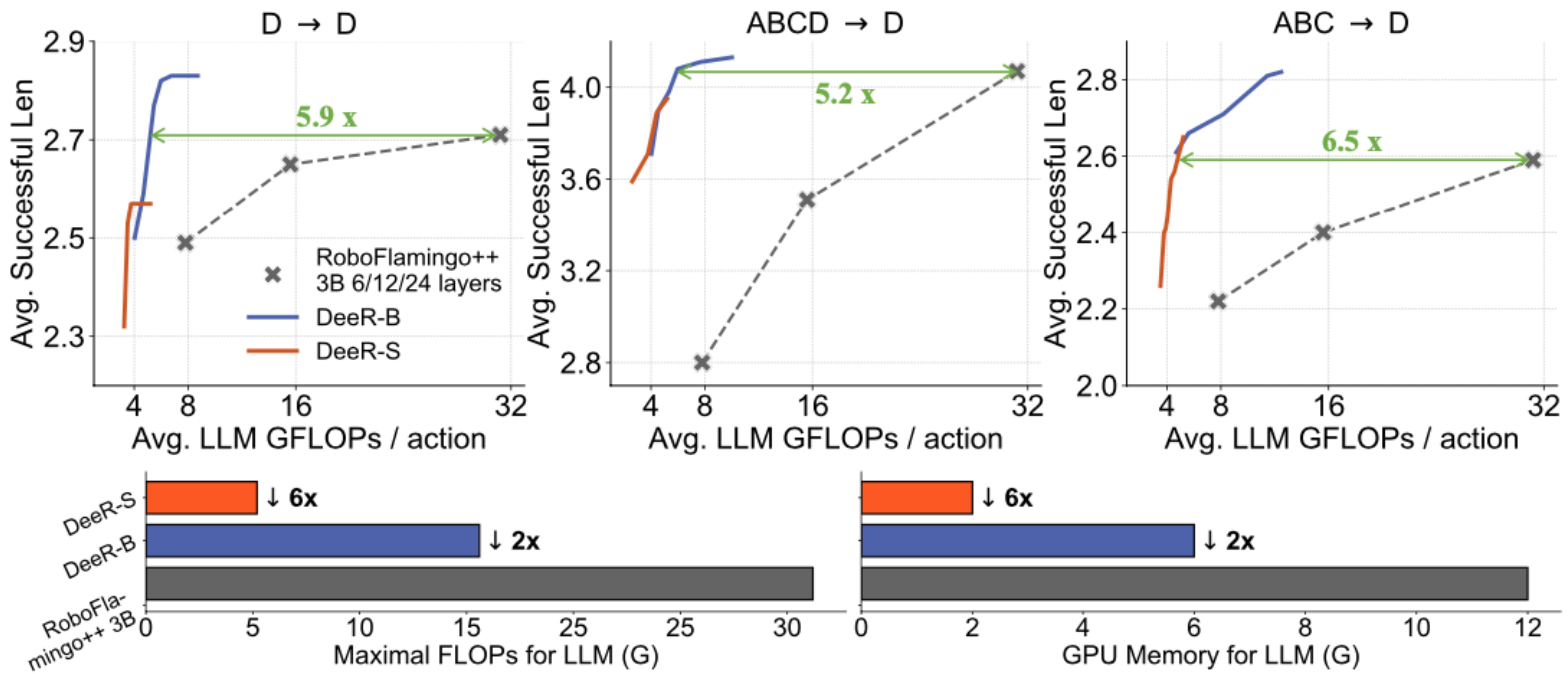
subject to

$$\text{FLOPs}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) < B,$$
$$\text{MFLOPs}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) < G,$$
$$\text{Mem}(\mathcal{T}, \{\eta_1, \eta_2, \ldots\}) < M.$$

Experiments

CALVIN Long-Horizon Multi-Task Language Control benchmark

- **5.2-6.4x** Computational Efficiency Improvement
- **2-6x** Reduction in Memory
- **No Loss** in Performance



Experiments

Tsinghua University

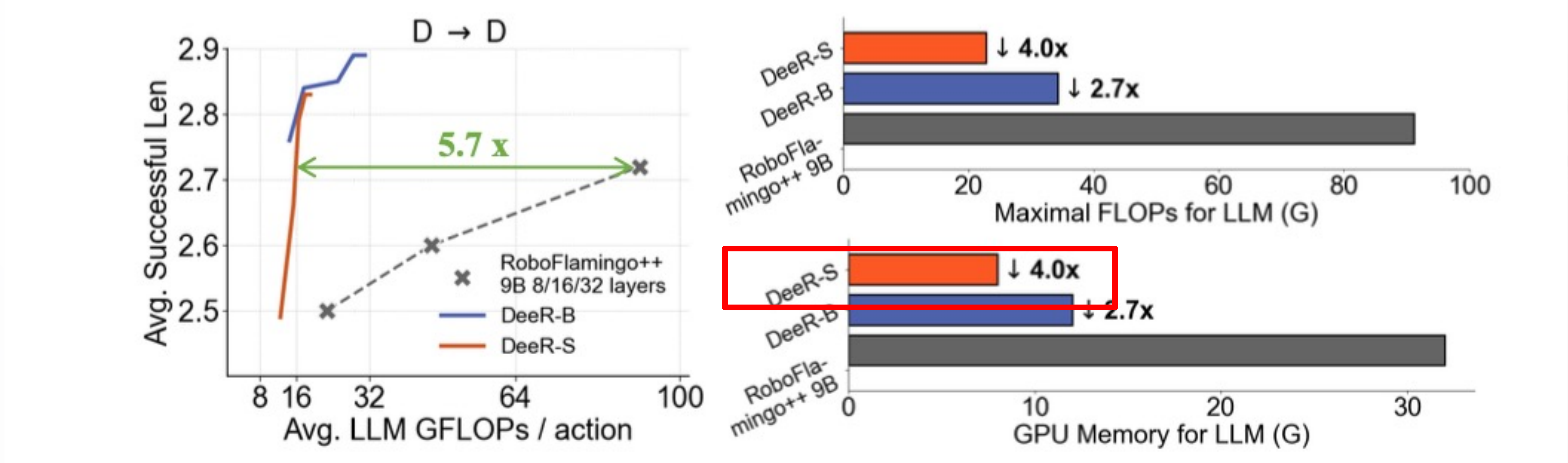
□ CALVIN Long-Horizon Multi-Task Language Control benchmark

Method	Input	Data	Foundation model	Avg. successful len (*LLM GFLOPs*) D→D	ABCD→D	ABC→D
GR-1 [74] (*ICLR'24*)	RGB+ Proprio	LANG	Video-pretrained Transformer	-	4.21	3.06
HULC [13] (*RA-L'22*)	RGB	ALL	✗	2.64	3.06	0.67
RT-1 [15] (*RSS'23*)	RGB	LANG	✗	-	2.45	0.9
SPIL [75] (*ICML'24*)	RGB	ALL	✗	2.67	-	1.71
SuSIE [76] (*ICLR'24*)	RGB	ALL	InstructPix2Pix [77]	-	-	2.69
RoboFlamingo (*ICLR'24*)	RGB	LANG	OpenFlamingo 3B	2.46 (*31.2*)	4.08 (*31.2*)	2.47 (*31.2*)
RoboFlamingo++	RGB	LANG	OpenFlamingo 3B	2.71 (*31.2*)	4.07 (*31.2*)	2.59 (*31.2*)
DeeR (ours)	RGB	LANG	OpenFlamingo 3B	**2.83** (*8.6*)	**4.13** (*10.0*)	**2.82** (*12.5*)
DeeR w. online (ours)	RGB	LANG	OpenFlamingo 3B	**2.92** (*8.5*)	**4.13** (*9.7*)	**2.90** (*9.5*)

Experiments

◻ CALVIN Long-Horizon Multi-Task Language Control benchmark

➢ Enabled Lossless Inference of a **9B** Embodied Large Model on an **8GB GPU**



Experiments

☐ More

- ➢ Real inference efficiency: **68.1% reduction** in LLM inference time
- ➢ DeeR is compatible with Quantization

Model	Len	GFLOPs	Time
Robo++	4.07	31.2	55ms
DeeR	4.08	6.0	17.5ms

DeeR	Memory	Avg Len
float32	6G	4.13
float16	3G	4.12
int4	1.7G	3.91

Take-away Message

- MLLMs failed to meet real-time requirements for robot
- There exists redundancy in MLLMs for embodied tasks
- DeeR adjusts MLLM size based on specific situations via early-exit
  - Multi-exit Architecture
  - Action Consistency for Early-Termination
  - Tailed sampling training strategy
- DeeR significantly reduces computational costs 5.2-6.4x and GPU memory usage 2x-6x